



Release Date: September, 2015

Updates:

Database Foundations

1-3

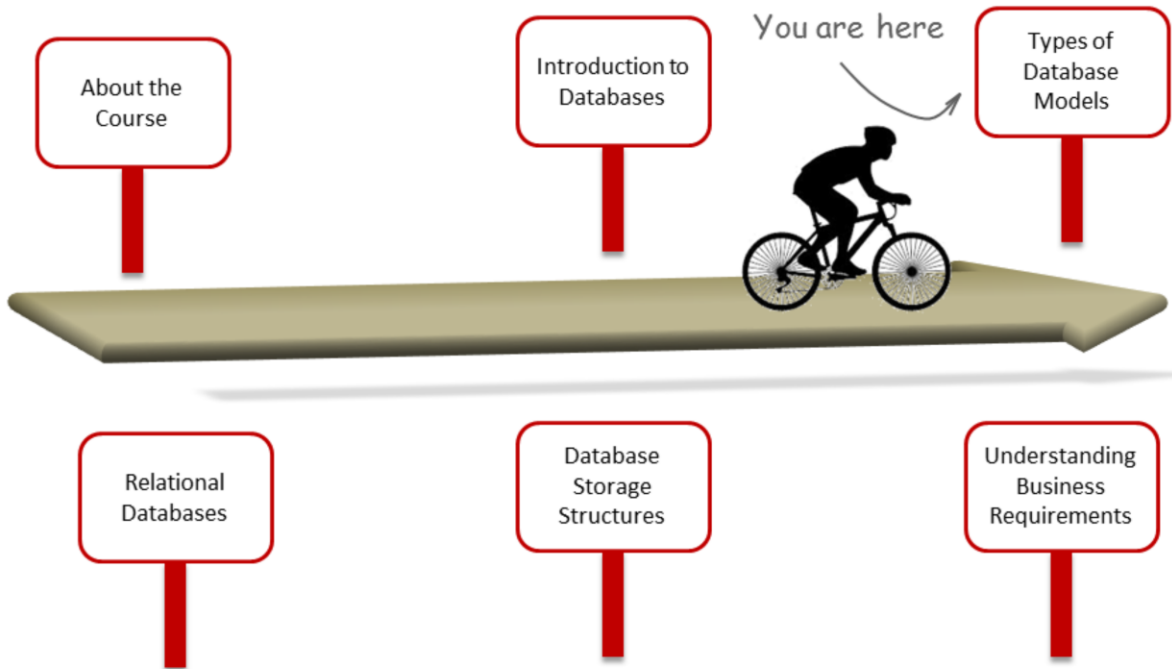
Types of Database Models



ORACLE ACADEMY

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Roadmap



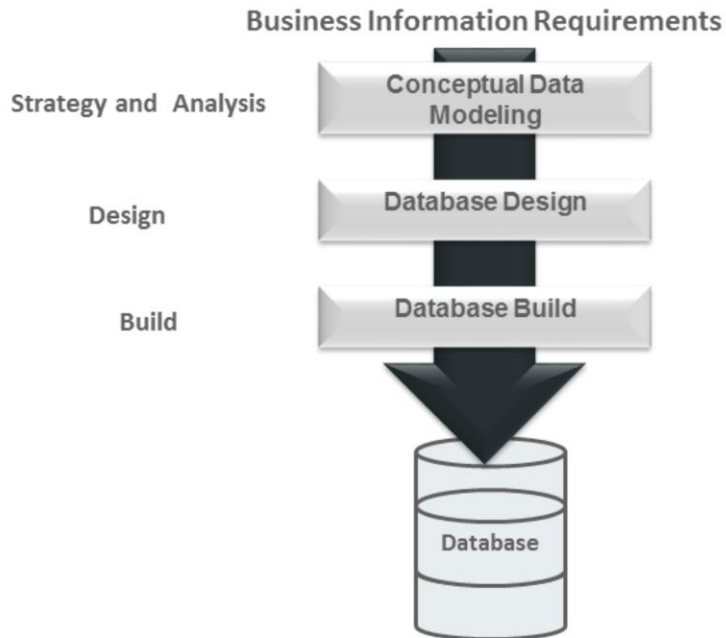
Objectives

This lesson covers the following objectives:

- Describe the database development process
- Explain the common types of database models:
 - Flat file model
 - Relational model
 - Hierarchical model
 - Network model
 - Object-oriented model



Database Development Process



Data modeling is the first part of the database development process. Conceptual data modeling is the examination of a business and business data to determine the structure of business information and the rules that govern it. This structure forms the basis for database design. A conceptual model is relatively stable over long periods of time. Physical data modeling (or database building) is concerned with implementation in a given technical software and hardware environment. The physical implementation is highly dependent on the current state of technology and is subject to change as available technologies rapidly change.

Common Types of Database Models

Flat File Model

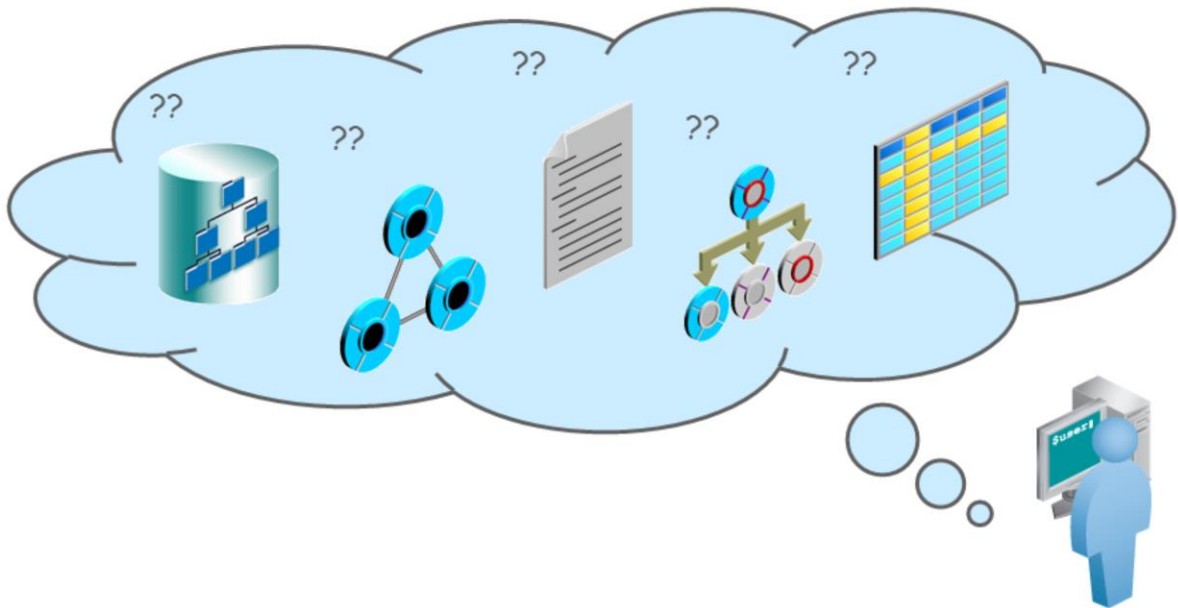
Hierarchical Model

Relational Model

Network Model

Object-Oriented Model

Case Scenario: Types of Database Models



This slide shows the student wondering about the different types of database models.

Flat File Model



	MEMBER_ID	FIRST_NAME	LAST_NAME	ADDRESS	CITY	ZIP
Record 1	CN0001	Velasquez	Carmen	283 King Street	Seattle	5876
Record 2	CN0002	Ngao	LaDoris	5 Modrany	Boston	5862
Record 3	CN0003	Nagayama	Midori	8 Centrale	Lagos	2544
Record 3	CN0003	Nagayama	Midori	9 Fox Town	Stowe	2544

A flat file database is a database designed around a single table. The flat file design puts all database information in one table, or list, with fields to represent all parameters. A flat file may contain many fields, often with duplicate data that is prone to data corruption. If you decide to merge data between two flat files, you need to copy and paste relevant information from one file to the other. There is no automation between flat files. If you have two or more flat files that contain client addresses, for example, and a client moved, you would have to manually modify the address parameters in each file that contains that client's information. Changing information in one file has no bearing on other files. Flat files offer the functionality to store information, manipulate fields, print or display formatted information and exchange information with others, through email and over the Internet. Some flat files may be attached to external files, such as text editors, to extend functionality and manage related information.

Flat file databases are generally in plain-text form, where each line holds only one record. The fields in the record are separated with delimiters, such as tabs and commas.

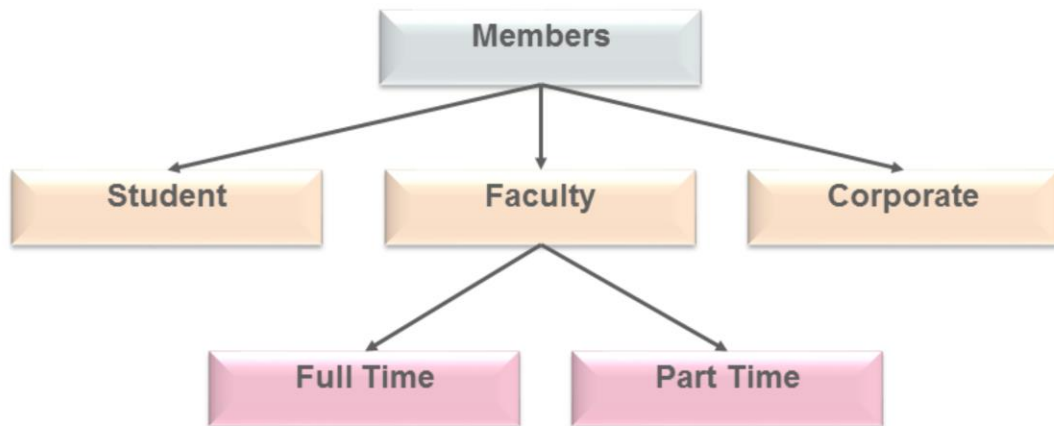
Example of a Flat File Model



	AUTHOR_ID	AUTHOR_NAME	TITLE
Record 1	AD0001	Oscar Wilde	A Vision
Record 2	AD0002	Leo Tolstoy	War and Peace
Record 3	AD0003	Oliver Goldsmith	Citizen of the World
Record 4	AD0003	Oliver Goldsmith	The Deserted Village

The slide depicts an example where the details of books and authors are stored in a single table. Notice that both the details of the book as well as authors are stored in this single table, and there is a repetition of values.

Hierarchical Model



In a hierarchical database model, the data is organized in a tree-like structure. The data is stored as records that are connected to one another through links. A record is a collection of fields. Each field contains only one value. The entity type of a record defines which fields the record contains.

A record in the hierarchical database model corresponds to a row in the relational database model. An entity type corresponds to a table.

In a hierarchical database model:

- Each child record has only one parent.
- A parent record can have one or more child records.

To retrieve data from a hierarchical database, the whole tree needs to be traversed, starting from the root node.

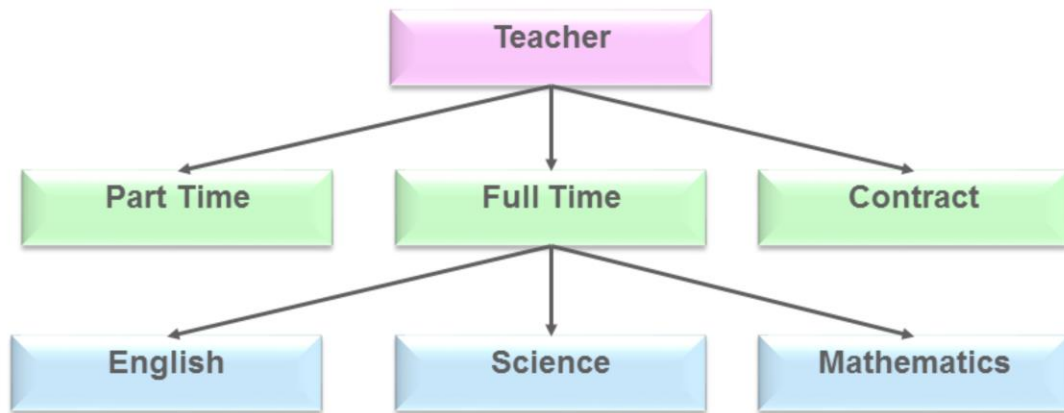
Advantages:

- Easy addition and deletion of new information
- Faster access to data at the top of the hierarchy

Disadvantages:

- Increased storage space
- Slower access to data at the bottom of the hierarchy

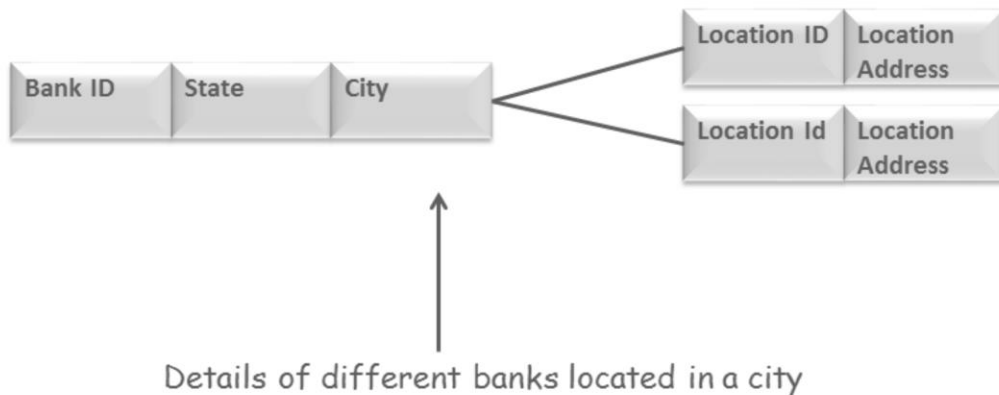
Example of a Hierarchical Model



The slide depicts an example of hierarchical model, where the root node represents the Teacher entity. A teacher could be employed as part time, full time, or contractor. Each parent node has child nodes. For example:

- The “Teacher” parent node has three child nodes that hold information about the type of employment.
- The “Full Time” parent node has three child nodes that hold information about the subjects taught by the teacher.

Network Model



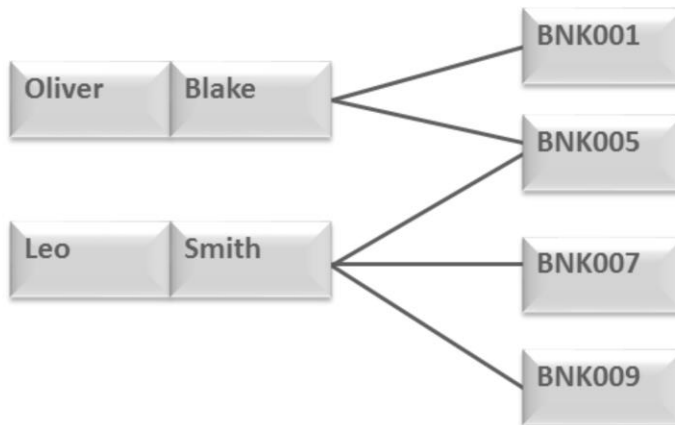
The network model is a database model that can be regarded as a flexible way of representing objects and their relationships. A network database comprised of a collection of records connected to one another through links. Each record is a collection of fields, each of which contains only one data value. A link is an association between two records.

In the Network Model Diagram as depicted in the slide:

- Boxes - correspond to fields, like Bank ID, State and City
- Lines - correspond to links that connect one record to another

In the hierarchical database model, data is represented as a tree of records, with each record having one parent record and many children. In a network database model each record can have multiple parent and child records, forming a generalized graph structure. The network model enables a more natural way of modeling the relationship between records.

Example of a Network Model



The slide depicts an example of a network model that stores information about bank account details of different people. In the example, Oliver Blake holds accounts in two banks, BNK001 and BNK005. Leo Smith holds accounts in three banks, BNK005, BNK007, BNK009. The records are connected to each other through links, represented by lines.

Relational Model

Table: STUDENT

The diagram shows a table with four columns and three rows. Red arrows point from the word 'Columns' to each of the four column headers: STUDENT_ID, NAME, ADDRESS, and BIRTH_DATE. Similarly, red arrows point from the word 'Rows' to each of the three data rows. The table is styled with a light blue header and a light blue body.

STUDENT_ID	NAME	ADDRESS	BIRTH_DATE
110	Jones	12 Oxford Street	03-Mar-1966
301	Smith	53 Hayes Drive	08-Dec-1953
134	Gonzales	5609 Maple Court	10-Feb-1987

ORACLE

ACADEMY

DFo 1-3
Types of Database Models

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. 14

A relational model describes a database in terms of tables, columns, rows, and joins between tables. Here are some key points about the relational model:

- Data is represented as a collection of tables.
- Each column represents attributes that belong to the table. For example, in the student table, you could have name, address, student ID, Birth_Date, and so on.
- Each row represents the instance of the table. For example, in the student table, 110, Jones, 12 Oxford Street, and 03-03-66 represents one student instance.
- Each table is the visual representation of columns and rows.
- Every table has a field or a set of fields that uniquely identifies the row. For example, in the student table, the student ID column uniquely identifies each student.

Features of a relational database model:

- The order of the rows and columns is not important.
- Every row is unique. There is a value in each row that is different from that value in another row.
- Each field can contain only one value.

- Values within a column of fields are from the same domain. (For example, a column defined as a date column would not contain a salary amount.)
- Table names must be unique, and column names within each table must be unique.

Example of a Relational Model

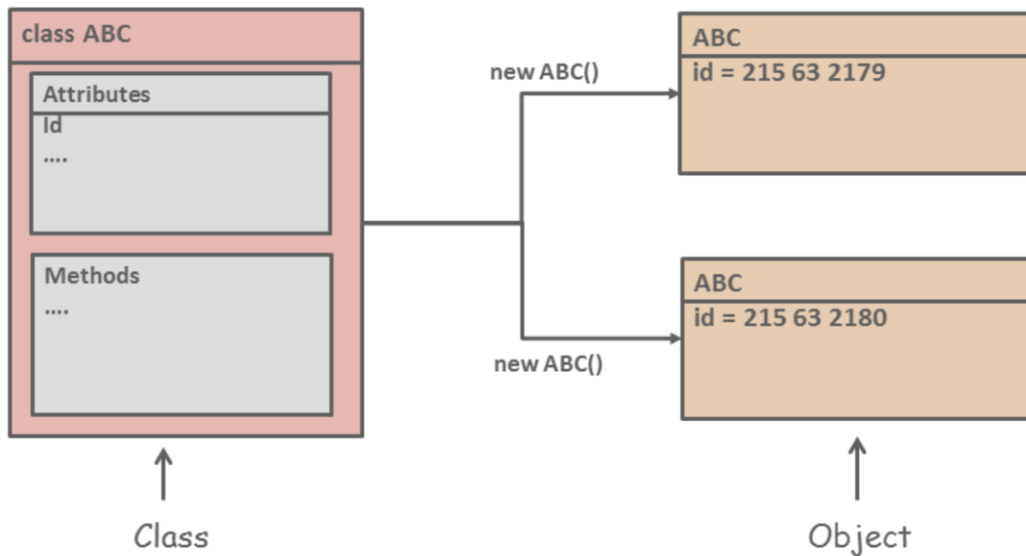
EMPLOYEE_ID	EMPLOYEE_NAME	HIRE_DATE	DEPARTMENT_ID
E0001	Jeff Covey	10-Jan-2003	10
E0002	William Jake	23-Nov-2006	20
E0003	Mary Schmidt	14-Jun-2007	30

A relational model represents a database as collections of records that are stored in tables. Each relational database table contains rows of records and columns with fields of information about each record.

Each table of records will have a relationship with another table of records when the two tables share a field (or column).

In the slide example, the Employees table contains a column that relates to the Department_ID from the Departments table. The inclusion of the Department_ID defines the relationship in the relational database model.

Object-Oriented Model



An object-oriented data model consists of the following basic object-oriented concepts:

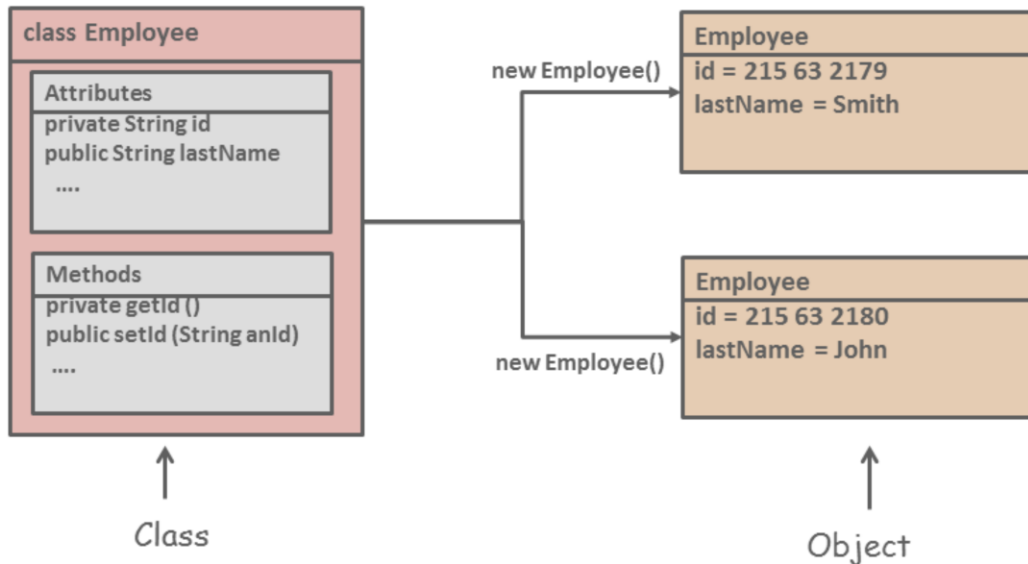
- A entity is modeled as an object.
- Every object has a state (the set of values for the attributes of the object) and a behavior (the set of methods that operate on the state of the object). The values and methods in an object can be accessed or invoked outside the object only through explicit implementation. The relationship between the objects is through sharing of access rather than through pointers or joins.
- An object must belong to only one class as an instance of that class.
- You can derive a new class (subclass) from an existing class (superclass).

Advantages

- Reduced maintenance
- Real-world modeling
- High code reusability

Disadvantages: Many information application systems do not benefit from object-oriented modeling because it is best suited for dynamic, interactive environments.

Example of an Object-Oriented Model



The graphic in the slide shows an Employee class defined with two attributes:

- The id attribute is the employee identifier.
- The lastName attribute is the last name of the employee.

The Employee class has two methods:

- getId()
- setId(String anId)

The id attribute and the getId() method are private and therefore can be accessed only within the class. The lastName attribute and the setId(String anId) method are public and can be accessed by other classes, too.

When you create an instance, the attributes store individual and private information relevant only to the employee. The information contained within an employee instance is known only to that particular employee. Each instance of Employee holds its own state. You can access that state only if the creator of the class defines it in a way that gives you access.

Summary

In this lesson, you should have learned how to:

- Describe the database development process
- Explain the common types of database models:
 - Flat file model
 - Relational model
 - Hierarchical model
 - Network model
 - Object-oriented model





ACADEMY